Naval Research Laboratory
Washington, DC 20375-5320

NRL/MR/6120--00-8480

Real-Time Probabilistic Neural Network Performance and Optimization for Fire Detection and Nuisance Alarm Rejection: Test Series 1 Results

SEAN J. HART
MARK H. HAMMOND
SUSAN L. ROSE-PEHRSSON
RONALD E. SHAFFER

Chemical Dynamics and Diagnostics Branch
Chemistry Division

DANIEL T. GOTTUK
MARK T. WRIGHT
JENNIFER T. WONG

Hughes Associates, Inc.
Baltimore, Maryland

THOMAS T. STREET
PATRICIA A. TATEM
FREDERICK W. WILLIAMS

Navy Technology Center for Safety and Survivability
Chemistry Division

August 31, 2000

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave Blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	August 31, 2000	Memorandum 2/1/00-5/3/00

4. TITLE AND SUBTITLE

Real-Time Probabilistic Neural Network Performance and Optimization for Fire Detection and Nuisance Alarm Rejection: Test Series 1 Results

5. FUNDING NUMBERS

PE — 603508N

6. AUTHOR(S)

Sean J. Hart, Mark H. Hammond, Susan L. Rose-Pehrsson, Ronald E. Shaffer, Daniel T. Gottuk,* Mark T. Wright,* Jennifer T. Wong,* Thomas T. Street, Patricia A. Tatem, and Frederick W. Williams

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Washington, DC 20375-5320

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/6120--00-8480

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
800 North Quincy Street
Arlington, VA 22217-5660

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

*Hughes Associates, Inc., Baltimore, MD

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (*Maximum 200 words*)

A series of tests were conducted to evaluate and improve the multivariate data analysis methods and candidate sensor suites used for the Early Warning Fire Detection (EWFD) system under development. The EWFD system is to provide reliable warning of actual fire conditions in less time with fewer nuisance alarms than commercially available smoke detection systems. Tests were conducted from 7-18 February 2000, onboard the ex-USS *Shadwell.* This report documents the performance of the probabilistic neural network achieved in real-time during this test series. Further optimization of the algorithm yielded performance gains over the real-time results. Simulation studies have been done to examine the effects of sensor drop-out, excessive noise, and erroneous sensor values.

14. SUBJECT TERMS

Multi-sensor	Multivariate analysis	Pattern recognition
Fire-detection	Smoke detector	Sensors
Sensor arrays	Probabilistic neural network	Sensor data fusion

15. NUMBER OF PAGES

39

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	UL

NSN 7540-01-280-5500

Standard Form 298 (Rev. 2-89)
Prescribed by ANSI Std 239-18
298-102

Table of Contents

1. Introduction

The U.S. Navy program Damage Control-Automation for Reduced Manning (DC-ARM), sponsored by the Office of Naval Research, PE0603508N, is focused on enhancing automation of ship fire and damage control systems. A key element to this objective is the improvement in situational awareness by improving the current fire detection systems. As in many applications, it is desirable to increase detection sensitivity, decrease the detection time and increase the reliability of the detection system through improved nuisance alarm immunity. Improved reliability is needed such that fire detection systems can provide quick, remote and automatic fire suppression capability. The use of multi-criteria based detection technology[1] offers the most promising means to achieve both improved sensitivity to real fires and reduced susceptibility to nuisance alarm sources. A multi-year effort to develop an early warning fire detection system is currently underway. The system being developed uses the output from sensors that measure different parameters of a developing fire or from analyzing multiple aspects of a given sensor output (e.g., rate of change as well as absolute value) and a neural network for fire recognition. A series of tests were conducted on the ex-USS SHADWELL[2] from February 7-18, 2000 to evaluate candidate prototypes of the early warning fire detection system (EWFD).

Improved fire recognition and low false alarm rates were observed using data from full-scale laboratory tests generated in a chamber located at Hughes Associates, Inc.[3,4,5] Several different sensor combinations were identified for use with a probabilistic neural network (PNN). Full-scale shipboard tests were conducted on the ex-USS SHADWELL to further develop detection algorithms and to expand the fire/nuisance source database.[6,7] Using these two data sets, two candidate suites of sensors were identified for prototype development.[7] Test Series 1 tested the real-time responses of the prototypes.[8] The algorithm development for the prototypes, the results of Test Series 1 shipboard testing, and the subsequent optimizations are described in this report.

The two data sets (laboratory and shipboard tests) served as the basis for a comprehensive PNN training data set used for the subsequent real-time applications. The classification of fire and nuisance events and the speed of the probabilistic neural network (PNN) were used to determine the performance of the multi-criteria fire detection system in Test Series 1. The EWFD system with the PNN developed for real-time detection, demonstrated faster response times to fires compared to commercial smoke detectors, while the overall classification performance was comparable to the commercial detectors. Some problems with the real-time implementation of the algorithm were identified and have been addressed. Using a variety of methods for speed and classification improvements, the PNN has been extensively tested and modified accordingly. As a result of the optimization efforts, significant improvements in

Manuscript approved July 25, 2000.

performance have been recognized. A detailed examination of PNN failures during fire testing has been undertaken and the initial results included. Using real data and simulated data, a variety of scenarios (taken from our recent field experiences) have been used or recreated for the purpose of understanding the behaviors and failure modes of the PNN in this application.

2. Theory and Real-Time PNN Code

The PNN is based upon Bayes' classification method.[9,10,11,12] The basis of the classification method is given in Equation 1, where h_i and h_j are the prior probabilities, c_i and c_j are the costs of misclassification, and $f_i(x)$ and $f_j(x)$ are the true probability density functions:

$$h_i\, c_i\, f_i\,(x) > h_j\, c_j\, f_j\,(x). \tag{1}$$

The difficulty with this relationship is that the prior probabilities (the probability that a sample will come from a given population distribution) are generally unknown and must be estimated from training data. This is done using Parzen's method of probability density function (PDF) estimation.[9] Bayes' classification will be more likely to group a new sample, x, into class i if the prior probability or the cost of misclassification is high. This is especially important for classifications where the cost of misclassification is not equal among the classes. In our application, false alarms, the potential cost of misclassifying a nuisance is much more serious than the cost for fire. In the PNN method different costs can be set for each class, thus producing a better classification for those classes that demand higher performance. Finally, if the probability density of a given class is large in the region of the new sample, x, then that class is favored. This allows for multi-modal distributions to be dealt with appropriately when a nearest neighbor-based classifier might fail. It has proven convenient and practical to implement the Parzen PDF estimator in a neural network format, the PNN.

The PNN is a nonlinear, nonparametric pattern recognition algorithm that operates by defining a PDF for each data class based on the training set data and the optimized kernel width parameter. For fire discrimination, the inputs are the sensor responses or pattern vectors. The outputs of the PNN are the Bayesian posterior probabilities (i.e., measures of confidence in the classification) that the input pattern vector is a member of one of the possible output classes, for example, fire or nuisance source.

The hidden layer of the PNN is the core of the algorithm. During the training phase, the pattern vectors in the training set are simply copied to the hidden layer of the PNN. Unlike other types of artificial neural networks, the basic PNN only has a single adjustable parameter. This parameter, termed sigma (σ), or kernel width, along with the members of the training set, define the PDF for each data class. In a PNN, each PDF is composed of Gaussian-shaped kernels of width σ located at each pattern vector. The PDF essentially determines the boundaries for classification. The kernel width is critical because it determines the amount of interpolation that occurs between adjacent pattern vectors. As the kernel width approaches zero, the PNN essentially reduces to a nearest neighbor classifier. A large kernel width has the advantage of producing a smooth PDF which exhibits good interpolation properties for predicting new pattern vectors. Small kernel widths reduce the amount of overlap between adjacent data classes. The optimized kernel width is a compromise between an overly small or large σ.

Prediction of new targets using a PNN is more complicated than the training step. Each member of the training set of pattern vectors (i.e., the patterns stored in the hidden layer of the PNN and their respective classifications), and the optimized kernel width are used during each prediction. As new pattern vectors are presented to the PNN for classification, they are serially propagated through the hidden layer by computing the Euclidean distance, d, between the new pattern and each pattern stored in the hidden layer. The Euclidean distance scores are then processed through a nonlinear transfer function (the Gaussian kernel) given in Equation 2:

$$\text{Hidden Neuron Output} = e^{\left(\frac{-(d)}{\sigma^2}\right)} \qquad (2).$$

Because each pattern in the hidden layer is used during each prediction, the execution speed of the PNN is considerably slower than some other algorithms. The mass data storage requirements can also be quite large since every pattern in the hidden layer is needed for prediction. Several researchers have developed modified PNN algorithms to overcome this limitation, but were not deemed necessary for this application.

3. Experimental

The selection of the sensors that comprise the two prototypes that were used during Test Series 1 was completed in December 1999. The laboratory and SHADWELL-collected data, which jointly comprised the PNN training set, have been described and discussed in great detail.[3,4,5] Furthermore, the down selection of sensors from a pool of 14 possible candidates has been described in detail.[7] The two sensor arrays chosen were: Prototype 1 – ionization (ION), photoelectric (Photo), carbon monoxide (CO), relative humidity (RH), and carbon dioxide (CO_2) and Prototype 2 – ION, Photo, CO, RH, and Temperature. The Simplex ionization and photoelectric sensors used in the laboratory and shipboard tests were not applicable to the real-time prototype. System Sensor ionization and photoelectric detectors were the best available substitute. Optimization experiments were performed prior to Test Series 1 in order to determine the other parameters of the PNN calculations to be performed in real-time. These optimization experiments included testing background subtracted, magnitude and slope calculation variations, and training set composition.

Real-Time PNN

The real-time Matlab code used during the Test Series 1 is given in Appendix A. In addition, a flowchart of the code is shown in Figure 1. This flowchart shows how the PNN is incorporated into the real time analysis of sensor data including pre-processing, pattern calculation and scaling. The vector of input sensor responses (Xcurrent), one number for each sensor in the array, comprise the set of data that is passed to the algorithm for pre-processing and PNN analysis during real-time deployment. For prototypes 1 and 2, the vector was 5 elements long, one for each sensor in the array. The input vector is passed to a buffer matrix (5x5) that is filled and an average is calculated over the rows. This was done to match the data collection of the ion and photo sensors with that of the other sensors during real-time analysis. After averaging, the data were pre-processed. Since raw sensor responses had been chosen, only the ion and photo detectors were manipulated. The conversion from ΔMIC to percent obscuration/ft and then from percent obscuration/ft to percent obscuration/m was performed for the ion sensor and the conversion from obscuration/ft to obscuration/m was

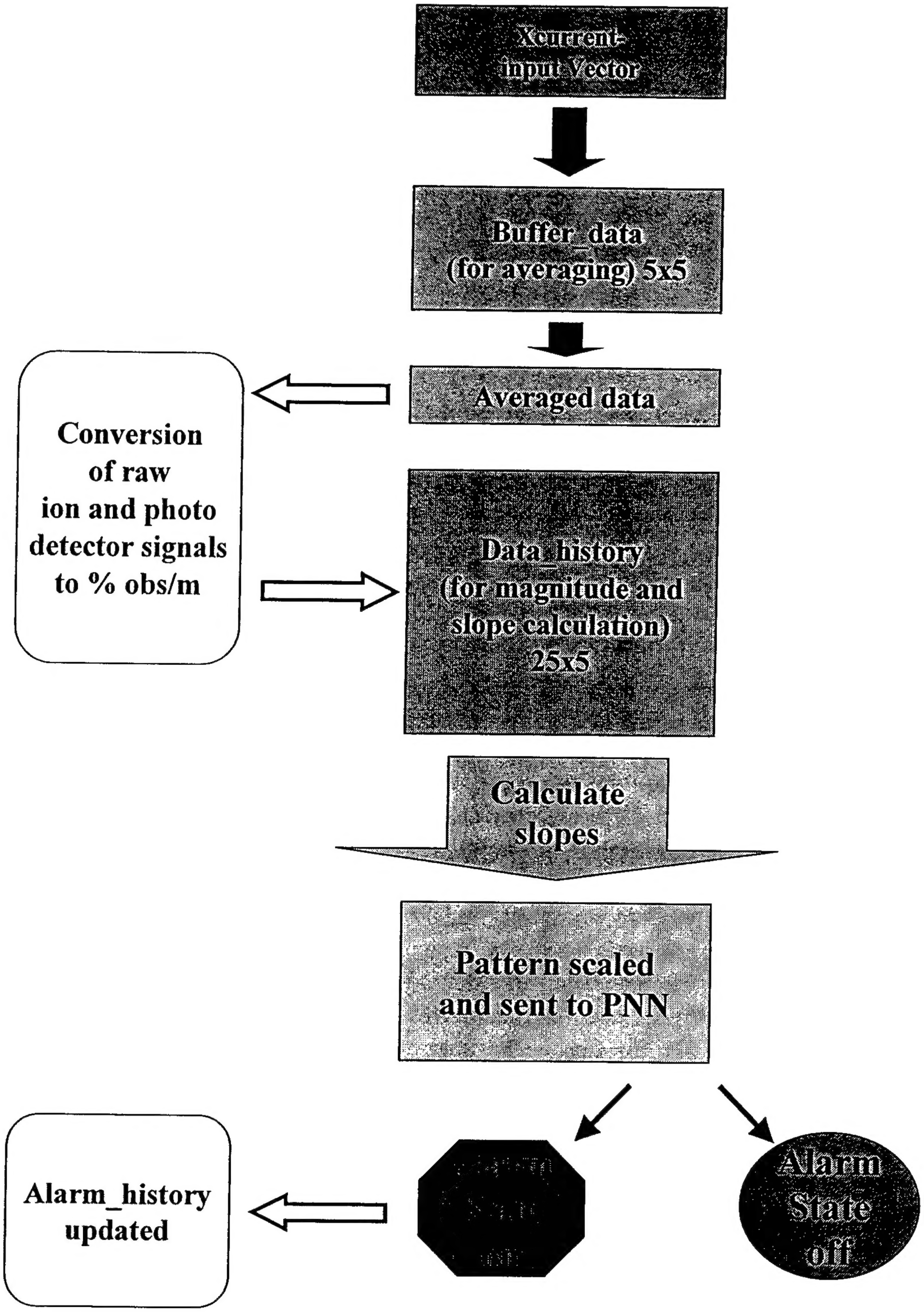

Figure 1. Flow chart for real-time PNN code. Buffer_data, data_history, and alarm_history are both inputs and outputs to code. All three variables have the most recent value added to the end of the vector or matrix and the first (oldest) value removed, thus maintaining their size.

done for the photo sensor. The resulting pattern (sensor vector) was added to the end of the 25x5 matrix, data_history, and the first row was deleted to maintain the size of the matix. In this manner new patterns were added and data_history was updated and reflects the most recent (25) patterns collected. From data_history, the pattern magnitudes and slopes were computed and then autoscaled (mean zero and unit variance) using the means and standard deviations derived from the training set. The resulting scaled pattern was then submitted to the PNN algorithm for the classification and determination of the probability of a fire event. The alarm state was triggered if the probability was greater than 0.75 for three consecutive predictions.

PNN Real-Time Implementation

The real-time deployment of the PNN combined data acquisition, processing and transfer of data to a Matlab script called from within Labview. The real-time Matlab code was written with several inputs that are also in the program's output list, updating variables that replace old data with new as the algorithm is successively called. The Labview program successfully called the PNN algorithm in real-time and classifications were made throughout the fire testing. The PNN performed well and generally produced results consistent with the commercial-off-the-shelf (COTS) sensors, manufactured by Simplex, Inc., which are installed in SHADWELL.[13] There were several occasions when the PNN alarmed faster than COTS for a fire and alarmed slower or did not alarm at all for a nuisance source.

There were several problem areas during the first real-time deployment of the PNN aboard the ex-USS SHADWELL. These include 1) a computational bottleneck that worsens as time increases, 2) high probability producing a false alarm shortly after the PNN calculation begins, 3) a calibration mismatch between the new Systems Sensors ionization detector (currently used in the prototypes) and the former Simplex ion detector (used during the collection of training data).

The first problem is characterized by the data acquisition system collecting data at a rate of 1 data point every 2 seconds, but after $\approx$ 2000 data points, the system required 6 seconds between collection of data points. This delta time represents a significant impediment to early fire warning and detection. The computational slowdown has been isolated and appears to be rooted in the data acquisition software's routines to call Matlab scripts. The instructions required for PNN analysis are contained in the Matlab scripts. While the cause of the slowdown has not been determined, other methods to streamline the interface and remove this bottleneck were investigated. Two general paths were considered: C or C++ code called from Labview and the PNN routines being programmed directly in Labview. The use of C code has been attempted without success due to difficulties with Labview calling the Matlab C/C++ function libraries. Further work will be done to make the current libraries functional or to find other more suitable library packages.

There were alarm spikes at the initiation of the PNN that occurred with the calculations performed in real time. In order to match the data in the training set as closely as possible, a 5-point (second) buffer was created to average the current System Sensor ion and photo detectors. This was done to match the current data acquisition with the Simplex system, which only produced new values every 5 seconds. The code error was that the mean of the matrix, data_average, was calculated even when buffer_data contained zeros. Even though the PNN was not called until data_history contained no zeros, the averaged values were included thus increasing the slope somewhat. For certain sensor combinations, namely

prototype 2, this caused an increase in probability that resulted in an alarm state. The increase was immediately after the PNN began calculation (after data_history was filled with non-zero values, 25 points). The solution was to remove the data-averaging step in the real time code, and this has remedied the problem. An additional problem occurred with the wiring on prototype 2 at location B. The temperature and humidity sensors (co-located in one housing by the same manufacturer) had their output wires crossed. This only became apparent during post processing when examination of the plots revealed features common to the relative humidity and the temperature sensor. For the purposes of prototype evaluation, units at location B were not used for comparison purposes in this report.

The most complex problem encountered was the mismatch of the System Sensor ionization data with that of the Simplex ion detectors. Originally it was thought that swapping one manufacturers' sensor for another brand would be possible, if both responses were converted to a standard such as percent obscuration per meter (%obs/m) as measured in UL Standard 268 smoke box sensitivity tests.. This turned out not to be the case, and an empirical correlation was required. Based on UL 268 smoke box tests conducted by System Sensor a general empirical correlation was established between the ΔMIC reading of the ion detector and the corresponding %obs/ft measurements in the smoke box. The fourth order polynomial used during the test series is shown in equation 3:

$$y(x) = 0.0000034x^4 - 0.0004140x^3 + 0.0171968x^2 - 0.2070225x + 0.0004794 \qquad (3)$$

Where y(x) is the % obscuration/ft and x is the System Sensor ionization detector ΔMIC reading. Plots showing the raw System Sensor data for several fires and the converted data are given in Figure 2. What is immediately noticeable is the change in the shape of the curves; the %obs/m data has a different temporal profile than the ΔMIC curve due to the non-linear correlation. Furthermore, the magnitudes of the curves do not agree with the magnitudes of the ion sensor used in the training set. The values for the ion sensor in the training set ranged between -2 and 12 compared with the System Sensor values that ranged between 0 and 90. This is due in part to the System Sensor detectors having a larger dynamic range than the Simplex detectors. This poor agreement between the System Sensor and Simplex detector outputs is likely to adversely affect the classification ability of the PNN. While the curve shape may compromise alarm time, the PNN still performs reasonably well due to the overlap of the number ranges (even though the System Sensor values can be much higher).

Test Series 1: Experiments and Sensor Combinations

Test Series 1 is described in Reference 8. The names, classifications, and descriptions of the tests performed are given in Table 1. Four prototypes were used. Data was also collected for oxygen (O_2), hydrogen sulfide (H_2S), nitrous oxide (NO), hydrocarbon (C1-C6), residential ion with top removed (R-ION Chamber), and residential ion (R-ION). As shown in Table 2, the sensor combinations were assigned numbers. Column numbers 2-6 are prototype 1a, 7-11 are prototype 2a, 12-16 are prototype 1b, 17-21 are prototype 2b, and 22-27 are the extra sensors.

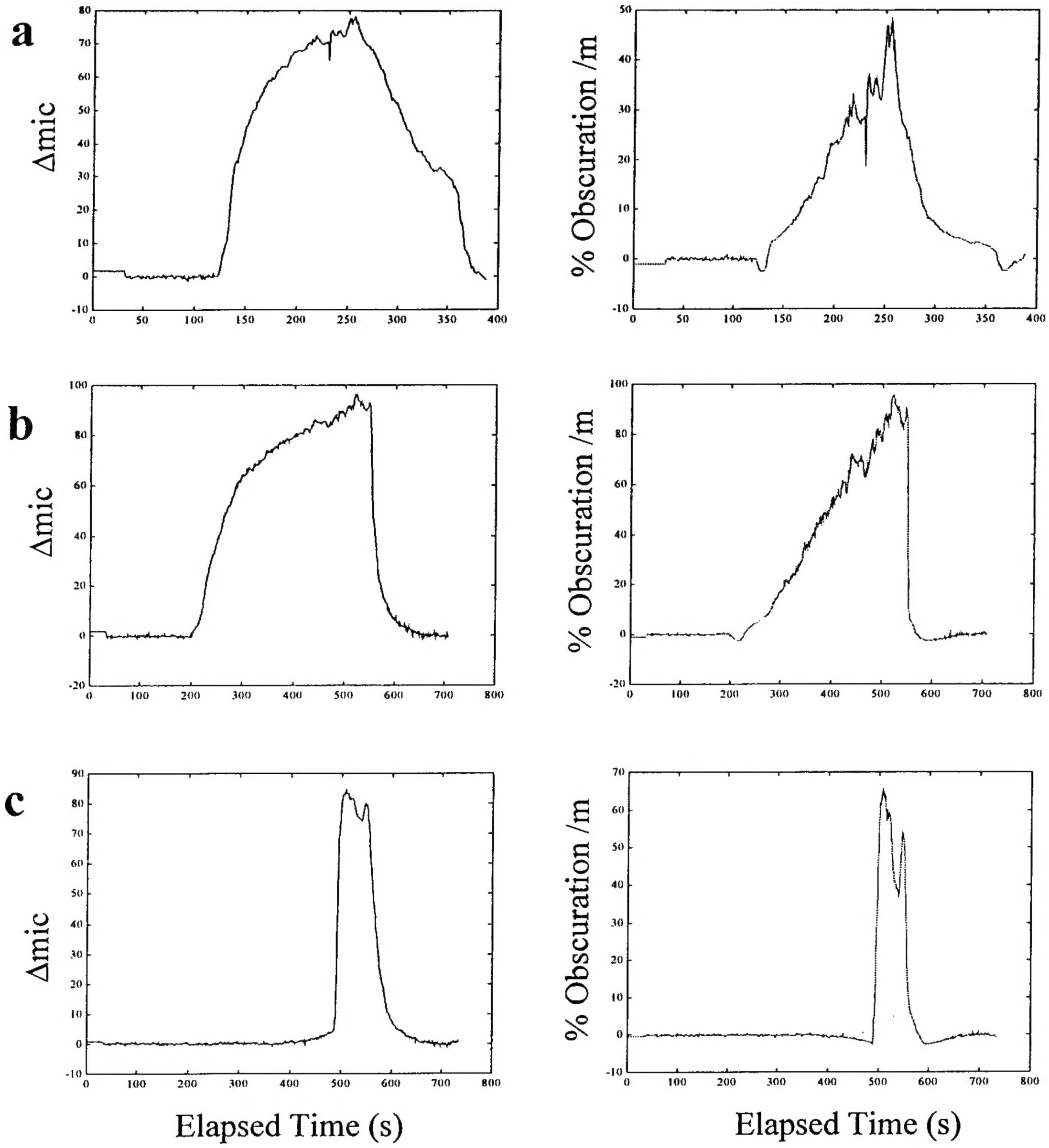

Figure 2. System Sensor ion detector output before and after conversion to % obscuration / m for several representative and duplicate fire sources: a) flaming heptane, b) flaming heptane, and c) oily rag in trashcan.

Table 1. Test name, type, classification (1 = fire, 2 = nuisance) and description

Test Name	Type	Class	Description
EWFD_001	fire, flaming	1	Heptane
EWFD_002	fire, flaming	1	Pipe insulation and fuel oil
EWFD_003	fire, flaming	1	Oily rag, newspaper, cardboard in sm. Trashcan
EWFD_004	nuisance/ fire	1	Burning toast
EWFD_005	fire, smoldering	1	Smoldering trash bag
EWFD_006	nuisance	2	Cigarette smoking
EWFD_007	fire, flaming	1	Flaming trashbag, TODCO wallboard
EWFD_008	fire, flaming	1	Heptane
EWFD_009	nuisance	2	Burning popcorn
EWFD_010	fire, flaming	1	Electrical cable and pipe insulation
EWFD_011	fire, smoldering	1	Smoldering electrical cable
EWFD_012	nuisance	2	Arc welding
EWFD_013	fire, flaming	1	Flaming bedding material
EWFD_014	fire, flaming	1	Oily rag, newspaper, cardboard in sm. Trashcan
EWFD_015	nuisance	2	Normal toasting
EWFD_016	fire, flaming	1	Small wood crib
EWFD_017	fire, flaming	1	Trashcan and office chair
EWFD_018	nuisance	2	Steel Cutting
EWFD_019	fire, smoldering	1	Smoldering bedding material
EWFD_020	fire, smoldering	1	Printed wire circuit board
EWFD_021	fire	1	Brief wire overheat
EWFD_022	fire, smoldering	1	Smoldering oily rag, newspaper, cardboard in sm. Trashcan
EWFD_023	fire, flaming	1	Pipe insulation and fuel oil
EWFD_024	Nuisance	2	Nylon rope
EWFD_025	nuisance/ fire	1	Nylon rope into sm. Trashcan
EWFD_026	fire, smoldering	1	Smoldering trash bag
EWFD_027	nuisance	2	Burning popcorn
EWFD_028	nuisance	2	Steel grinding
EWFD_029	fire, smoldering	1	Smoldering bedding material
EWFD_030	nuisance/ fire	1	Burning toast
EWFD_031	fire, flaming	1	Pipe insulation and heptane
EWFD_032	nuisance	2	Cigarette smoking
EWFD_033	fire, smoldering	1	Printed wire circuit board
EWFD_034	fire	1	Brief wire overheat
EWFD_035	fire, smoldering	1	Smoldering oily rag, newspaper, cardboard in sm. Trashcan

Table 2. Sensor number and identity. Column numbers 2-6 are prototype 1a, 7-11 are prototype 2a, 12-16 are prototype 1b, 17-21 are prototype 2b, and 22-27 are the extra sensors

Sensor	Column #
Elapsed time	1
ION#1(MIC)	2
PHOTO#1(%/ft)	3
CO(ppm)	4
RH(%)	5
CO2(ppm)	6
ION#4(MIC)	7
PHOTO#4(%/ft)	8
CO(ppm)	9
RH(%)	10
Temp(°C)	11
ION#2(MIC)	12
PHOTO#2(%/ft)	13
CO(ppm)	14
RH(%)	15
CO2(ppm)	16
ION#3(MIC)	17
PHOTO#3(%/ft)	18
CO(ppm)	19
RH(%)	20
Temp(°C)	21
O2(%)	22
H2S(ppm)	23
NO(ppm)	24
C1-C6(ppm)	25
RION Chamber (V)	26
RION(V)	27

4. Preparation / Optimization Results for Test Series 1

A priori PNN training and optimization was conducted before the first deployment of the PNN in real-time aboard the SHADWELL in Test Series 1 and were used to determine the optimum set of preprocessing parameters to use in real time. The parameters investigated included background subtraction, and various slope / magnitude (mag) determinations. These tests were determined using a training set consisting of the combined laboratory and shipboard tests. A small set of laboratory and shipboard data was removed and used as a prediction set for validation purposes.

The effects of preprocessing on the total classification rate are shown in Table 3. Three preprocessing (preproc) methods were used. All of the data, except the smoke detectors, were in engineering units when received from the data acquisition system. The first method used background subtraction. Empirical correlations based on UL 268 smoke box tests were used to convert the smoke detector output readings to units of percent obscuration per meter. These correlations are based on the calculation of I_o, which is essentially the average background reading. For both the gas sensors and the smoke detectors, the first 60 seconds of each test was used to determine the baseline reading. Data collected 30 seconds prior to source initiation were used to compute the background pattern for the laboratory data. The average of the first 30 seconds of background was used for the shipboard data. The second method used all the data as received, referred to as raw. The third method used the gas sensor as received in engineering units, while the smoke detectors were converted to % obs/m. The best results were obtained with methods 2 and 3. While raw data with magnitudes only produced the highest classification percentage (88%), this method did not allow for conversion to a common standard for the Simplex and System Sensor detectors. Therefore, the most desirable result used method 3, selection 8 (refer to Table 3) and yielded a classification of 85% correct. Both magnitudes and slopes, with the longest slope length calculation (25 points), were selected for use during Test Series 1.

Further optimization experiments investigated the use of various combinations of training set pattern alarm times. The single best alarm time was 11% with 85% total correct classification. An earlier report[4] indicated that the use of multiple alarm times significantly improved the ability of the PNN to remain in alarm. However, in this case the overall classification results using multiple alarm times were lower as shown in Table 4.

Table 3. Preprocessing experiments and total % correct achieved using prototype 1, at the 11% alarm time criteria

Experiment	% Correct
preproc = 1 - background subtraction (relative ppm) and obs./m	
1 magnitude(5)	80.0
2 magnitude(10)	75.0
3 magnitude(5) + slope(10)	77.5
4 magnitude(10) + slope(10)	77.5
5 magnitude(5) + slope(15)/slope(15)	75.0
6 magnitude(10) + slope(15)/slope(15)	77.5
7 magnitude(5) + slope(25)/slope(15)	77.5
8 magnitude(10) + slope(25)/slope(15)	77.5
preproc = 2 - raw	
1 magnitude(5)	87.5
2 magnitude(10)	85.0
3 magnitude(5) + slope(10)	77.5
4 magnitude(10) + slope(10)	77.5
5 magnitude(5) + slope(15)/slope(15)	77.5
6 magnitude(10) + slope(15)/slope(15)	80.0
7 magnitude(5) + slope(25)/slope(15)	82.5
8 magnitude(10) + slope(25)/slope(15)	82.5
preproc = 3 - ppm and obs/m	
1 magnitude(5)	82.5
2 magnitude(10)	82.5
3 magnitude(5) + slope(10)	80.0
4 magnitude(10) + slope(10)	77.5
5 magnitude(5) + slope(15)/slope(15)	80.0
6 magnitude(10) + slope(15)/slope(15)	77.5
7 magnitude(5) + slope(25)/slope(15)	85.0
8 magnitude(10) + slope(25)/slope(15)	85.0

Table 4. PNN-CV total percent correct at various alarm time combinations for prototype 1, using preprocessing type 3 and magnitude/slope selection 8 (refer to Table 3 for definitions).

Alarm Time	% Correct
1	85.0
2	77.5
3	75.0
1 2	77.5
2 3	75.0
1 3	75.0
1 2 3	77.5

Alarm time thresholds (%obs/m for photo)

1	11.00%
2	1.63%
3	0.82%

5. PNN Real-Time Results

The results of the real-time deployment of the PNN onboard the SHADWELL in Test Series 1 are encouraging. The comparison of the real-time prototype performance with the COTS units is given in Table 5. The 77% overall correct classification of prototype 1 is similar to the COTS photo. The fire detection rate for both prototypes 1 and 2 was greater than both the COTS photo and ion sensors. Prototype 1 had better real-time performance than the COTS for fire detection, 89% of fires correctly classified. Prototype 2 correctly classified 85% of the fire scenarios whereas the COTS photo sensor correctly classified 81% of the fires. The nuisance performance for prototypes 1 and 2 was poorer than the COTS with 44 % correct classification versus 56% for the COTS ion sensor and 67% for the COTS photo sensor. If one considers the combined performance of the COTS photo and ion detectors (i.e., OR logic, where either one alarming constitutes an alarm), the fire detection is slightly better, but the nuisance rejection suffers. In many cases, the prototypes were faster to alarm for fires than either COTS unit. For example, using prototype 1, there were 8 fires faster than ion (FFTI) and 17 fires similar to ion (FSimTI). Fires or nuisances that are categorized as similar (Sim) to ion or photo are within ± 30 seconds of each other. This

means that for 25/26 fires (98%) this prototype was as fast or faster than the COTS ion detector. Similarly, for nuisance sources, several prototypes were slower to alarm than the COTS units and for several other experiments the alarm times were similar.

Table 5. Results of PNN classification during Test Series 1 for real-time execution (laboratory and 1999 shipboard tests used for training set)

	Sensors	Total % Correct	Fires % Correct	Nuisances % Correct	# Fires Correct (26)	# Nuisances Correct (9)
Prototype 1:	2 3 4 5 6	77.1	88.5	44.4	23	4
Prototype 2:	7 8 9 10 11	74.3	84.6	44.4	22	4
COTS	Ion	65.7	69.2	55.6	18	5
	Photo	77.1	80.8	66.7	21	6
	Ion or Photo	77.1	92.3	33.3	24	3

	FFTI	FFTP	FSimTI	FSimTP	NSTI	NSTP	NSimTI	NsimTP	TFF	TNS	TFSim	TNSim
Prototype 1:	8	14	17	8	3	1	3	4	4	0	17	5
Prototype 2:	8	15	15	7	3	1	3	4	7	0	12	5

FFTI: number of fires faster than ion
FFTP: number of fires faster than photo
FsimTI: number of fires similar to ion
FsimTP: number of fires similar to photo
NSTI: number of nuisances slower than ion
NSTP: number of nuisances slower than photo
NsimTI: number of nuisances similar to Ion
NsimTP: number of nuisances similar to photo
TFF: total number of fires faster than both ion and photo
TNS: total number of nuisances slower than both ion and photo
TFSim: number of fires similar to COTS
TNSim: number of nuisances similar to COTS

6. PNN Optimization

The PNN was modified and optimized for increased nuisance rejection relative to its real-time performance during Test Series 1. The optimization was done in several ways beginning with the correction of the real time code described in the previous section. The real time code used for optimization was identical to the modified code (used in the Test Series 2) with the exception of provisions (if statements) for proper preprocessing of various sensor values. This was done in order to allow testing of alternative sensor combinations. Similar to the algorithm for real-time analysis, the magnitudes and slopes were calculated

over a 25-point region (number of rows in data_history), no background subtraction was performed, and the System Sensor ion detector had a calibration conversion from ΔMIC to % obscuration/ft to % obscuration/m, and photo detector had a conversion from % obscuration/ft to % obscuration/m. Based upon the training set results given in Table 4, the 11% alarm level was used to generate the patterns, due to its higher classification performance.

The other main type of optimization was the modification of the training set. When developing a training set that must withstand the rigors of time and changing conditions, much care should be exercised to ensure that the data in the training set be as numerically and experimentally compatible with the prediction data to be classified. This is a continuous process of training set renewal / regeneration, and calibrations should be used if possible to determine when and which parts of the training set need to be updated. Following Test Series 1, the inclusion of propane as a fire sources was questioned because it is very small and if present onboard ship would be used in a controlled condition. Additional changes included the removal of 146 background patterns, which were thought to bias the data with respect to background signals. All the data were reviewed for quality and experimental integrity, as poor data in the training set will lead to lower classification performance.

Many patterns were removed based upon data quality and knowledge of the anticipated shipboard scenarios. Many of the backgrounds in both the laboratory and shipboard data sets that comprise the training set were removed. One entire class of fires, propane fires, were removed due to their small sensor response, which is similar to many nuisances. The experiment patterns that were removed are listed in Table 6. This reduced the number of samples in the training set from 327 to 170. The reduced size of the training set should improve both the calculation speed, due to the reduced size, and the performance, neglecting poor training examples. In an effort to test and characterize the training set and other variable combinations, 500 experiments were performed using code that calls the nearly identical code used in real time. The experiments varied 50 sensor combinations, 2 training sets (new and old), and 5 training set alarm conditions.

Table 6. List of experiment patterns removed from original "old" 327 pattern training set. None of the 29 Shadwell fires were removed. Meker burner and Bunsen burner represent propane fires.

Matlab Matrix Index	Report Name	Description
Laboratory Fires (11):		
8-11	com020 - com023	Meker burner w/marinite and vertical
17	com029	Bunsen burner
40	com055	Smoldering pillow
46	com063	Meker Burner, horizontal
54-56	com75 - com77	Burning toast, one slice
91	com116	Meker burner
Laboratory Backgrounds(90):		
126-160	-	Background
166-180	-	Background
186-200	-	Background
206-220	-	Background
226-235	-	Background
Shadwell backgrounds (56):		
271-296	-	Background
298-327	-	Background

Classification Performance

A representative set of classification results using various sensor combinations and training sets, is given in Table 7. The two prototype responses for the old and new training set are included for comparison. The use of the new training set with prototype 1 and 2 had a beneficial impact on the nuisance classification.

Table 7. Representative PNN classification results of optimization of sensor combinations and training sets

	Sensors	Total % Correct	Fires % Correct	Nuisances % Correct	# Fires Correct	# Nuisances Correct	FFTI	FFTP	FSimTI	FSimTP	NSTI	NSTP	NSimTI	NSimTP	TFF	TNS	TFSim	TNSim
Prototype 1:																		
327 rows "old", 11%	2 3 4 5 6	77.1	88.5	44.4	23	4	8	14	17	8	3	1	3	4	4	0	17	5
170 rows "new", 11%	2 3 4 5 6	82.9	88.5	66.7	23	6	7	13	16	8	4	3	4	4	3	1	15	6
Prototype 2:																		
327 rows "old", 11%	7 8 9 10 11	74.3	84.6	44.4	22	4	8	15	15	7	3	1	3	4	7	0	12	5
170 rows "new", 11%	7 8 9 10 11	74.3	80.8	55.6	21	5	5	12	17	9	3	2	4	4	3	0	15	6
Others:																		
170 rows "new",0.8%	27 4 6	74.3	84.6	44.4	22	4	10	12	14	10	3	3	5	4	5	1	15	6
170 rows "new", 11%	2 3 5 6 22	80.0	88.5	55.6	23	5	9	16	17	7	2	2	3	2	8	1	15	3
170 rows "new",1.6%	2 3 4 5 22	80.0	88.5	55.6	23	5	8	10	14	11	3	3	5	3	2	1	15	5
170 rows "new",0.8%	4 5 6 22 25	74.3	84.6	44.4	22	4	7	14	15	7	2	2	2	2	5	1	13	3
170 rows "new",1.6%	3 4 5 6	77.1	88.5	44.4	23	4	7	12	13	8	2	3	4	2	3	1	12	4
170 rows "new",1.6%	2 3 4 5 11 22	80.0	88.5	55.6	23	5	8	11	13	10	2	3	6	3	2	1	14	5
170 rows "new",1.6%	2 3 4 5 6 22	80.0	92.3	44.4	24	4	9	16	16	6	3	1	3	4	6	0	15	5
COTS	Ion	65.7	69.2	55.6	18	5												
	Photo	77.1	80.8	66.7	21	6												
	Ion or Photo	77.1	92.3	33.3	24	3												

FFTI: number of fires faster than ion
FFTP: number of fires faster than photo
NSTI: number of nuisances slower than ion
NSTP: number of nuisances slower than photo
TFSim: number of fires similar to COTS
Number of fires = 26; number of nuisances = 9
Similar to COTS (ion or photo) is defined as ± 30 seconds

TFF: total number of fires faster than both ion and photo
TNS: total number of nuisances slower than both ion and photo
FsimTI: number of fires similar to ion
NsimTP: number of nuisances similar to photo
TNSim: number of nuisances similar to COTS

The classification results using prototypes 1 and 2 using the new optimized training set with 170 patterns showed marked improvement especially with regards to nuisances. Prototype 1 with the new training set correctly identified 6 of the 9 nuisance sources. This is comparable to the performance observed for the COTS photo. For prototype 2 with the new training set, the overall performance remained the same, however the number of nuisance sources correctly identified increased. If one considers the combined response of both the ion or photo sensor, the nuisance rejection is much worse (33% correct) while the fire detection is better (92 % correct). This is due to the increased opportunity to alarm during a fire but also to false alarm during a nuisance event.

The results of several other high performing sensor combinations are also given in Table 7. An example from the battery of optimization experiments performed is the combination of R-ION, CO and CO_2 (i.e., 27, 4, 6). This combination of sensors produced a reasonable nuisance classification rate while maintaining a good level of fire detection and fast response. The % fire correct was 85, lower than the performance of prototype 1 and better than or equal to that of prototype 2: 81% - 85% correct. The nuisance % correct was 44.4, which was lower than the best COTS or other PNN-prototype nuisance classification rate. There are several additional factors, which make the R-ION sensor attractive. These include its low cost, availability, and most importantly the availability of consistent training data. There were several other top performing sensor combinations that had a nuisance classification rate of 56% and a fire detection rate of 89%; these combinations all contained O_2.

Speed Performance

In addition to the classification performance, the speed of PNN classification relative to COTS detectors was gauged using several criteria. The additional criteria used were the number of fires faster than ion (FFTI), number of fires faster than photo (FFTP), number of nuisances slower than ion (NSTI), number of nuisances slower than photo (NSTP), total number of fires faster than both ion and photo (TFF), and total number of nuisances slower than both ion and photo (TNS). The number of fires similar to ion (FsimTI) and the number of nuisances similar to photo (NsimTP) are determined by counting those experiments where the prototype response is within ± 30 seconds of the COTS response time. The number of fires similar to COTS (TFSim) and the number of nuisances similar to COTS (TNSim) are additional measures with which to determine PNN performance. These figures of merit results are given in Table 7.

For both prototypes, there were between 12-15 fires faster than photo and 5-8 fires faster than ion. Conversely, for both prototypes, there were between 1-3 nuisances slower than photo and 3-4 nuisances slower than ion. For both prototypes there were 15-17 experiments where the alarm times were similar to ion sensors, and 7-9 experiments where the alarm times were similar to photo sensors. Any number of fires or nuisances faster or slower than COTS represents an improvement obtained using the prototype and PNN. The low nuisance classification rate of the combined COTS detectors (ion and photo) indicates that they should not be used simultaneously for classification. It is

therefore unlikely to be useful to judge the PNN relative to the combined response times of the COTS sensors. For prototypes 1 and 2, the FFTI was 5-8 and the FsimTI was 15-17; this means that the prototypes and PNN responded the same or faster than the COTS ion unit for 77% - 96% of the experiments. For prototypes 1 and 2, the FFTP was 12-15 and the FsimTP was 7-9; this means that the prototypes and PNN responded the same or faster than the COTS photo unit for 73% - 92% of the experiments.

The third prototype, R-ION, CO, CO2, had a higher number of FFTI, 10, compared with 5-8 for the other prototypes. The FFTP, 12, was similar compared with 12-15 for the other prototypes. This prototype responded the same or faster than the COTS ion unit for 92% of the experiments and the same or faster than the COTS photo unit for 85% of the experiments.

Probability Plots

The probability plots for all 35 experiments performed during Test Series 1 using the a) old - 327 pattern training set and prototype 1, and the b) new - 170 pattern training set and prototype 1, and the c) sensor combination [R-ION, CO, CO2] with the new - 170 pattern training set generated at the 0.82% alarm time, can be seen in Figures 3 - 7 a b c respectively. Comparing the plots shows the effect of the various training sets / sensor combinations. When comparing the new training set and the old one for prototype 1 (Figures 3-7 a, b), the biggest differences are observed in tests 5, 9, 12, 17, 18, 27 and 30. Most of these tests are nuisance sources that did not alarm or the alarm was significantly delayed by using the new training set. In test 5, smoldering trash, an earlier alarm occurred using the new training set. In one case, test 17, trash can/office chair fire, the alarm was delayed by the new training set. It is also interesting to note that the baseline probability was elevated for tests 22-35. The relative humidity was elevated during those tests; therefore the new training set may be more susceptible to changes in the ambient conditions.

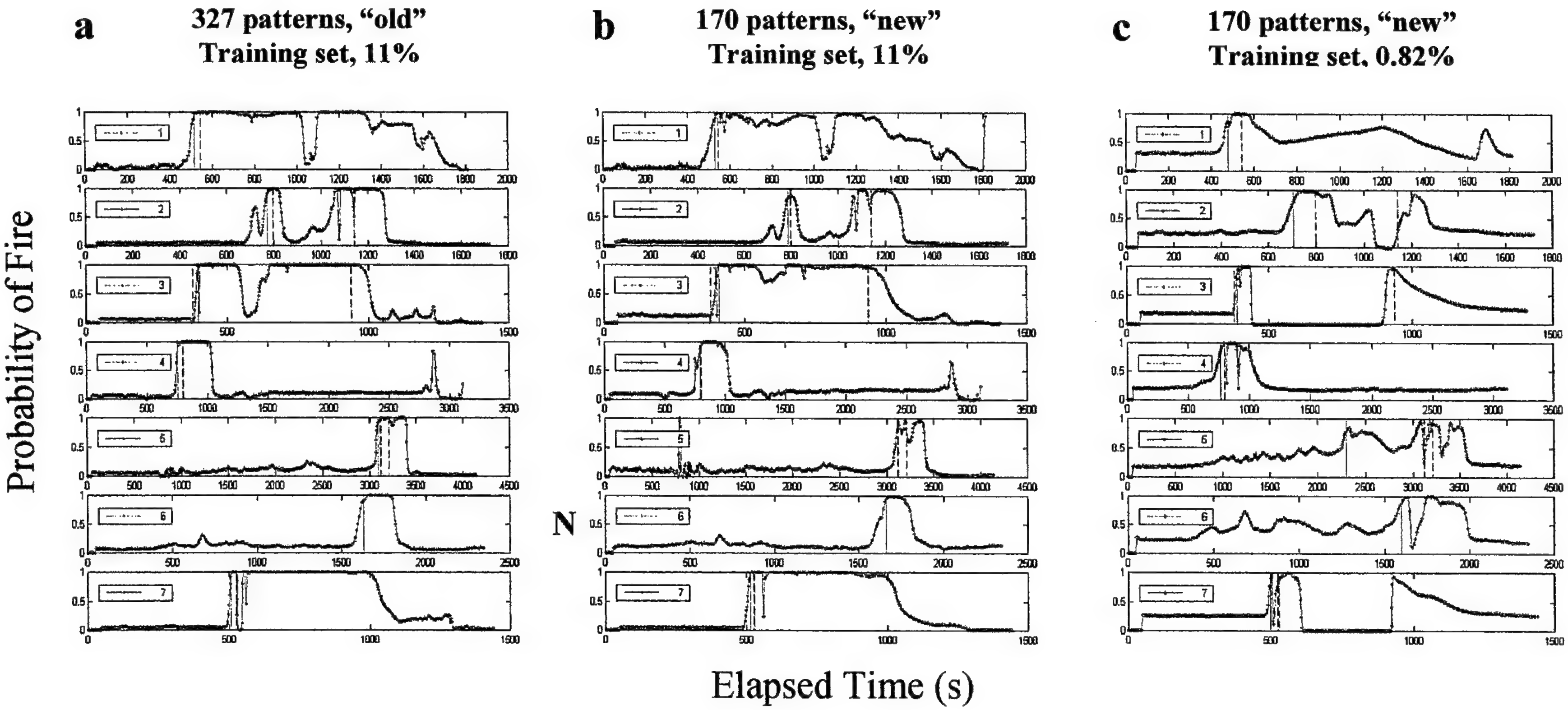

Figure 3. Probability versus elapsed time for Test Series 1 experiments 1-7 (refer to Table 1 for test descriptions) comparing a) the old, 327 pattern training set, generated at the 11 % alarm time, with prototype 1, b) the new, 170 pattern training set, generated at the 11 % alarm time, with prototype 1, and c) the sensor combination [R-ion, CO CO_2] using the new training set generated at the 0.82 % alarm time. Nuisance experiments are denoted with an "N". Solid green line = PNN alarm time, dashed red line = COTS Ion alarm time, and dashed black line = COTS Photo alarm time.

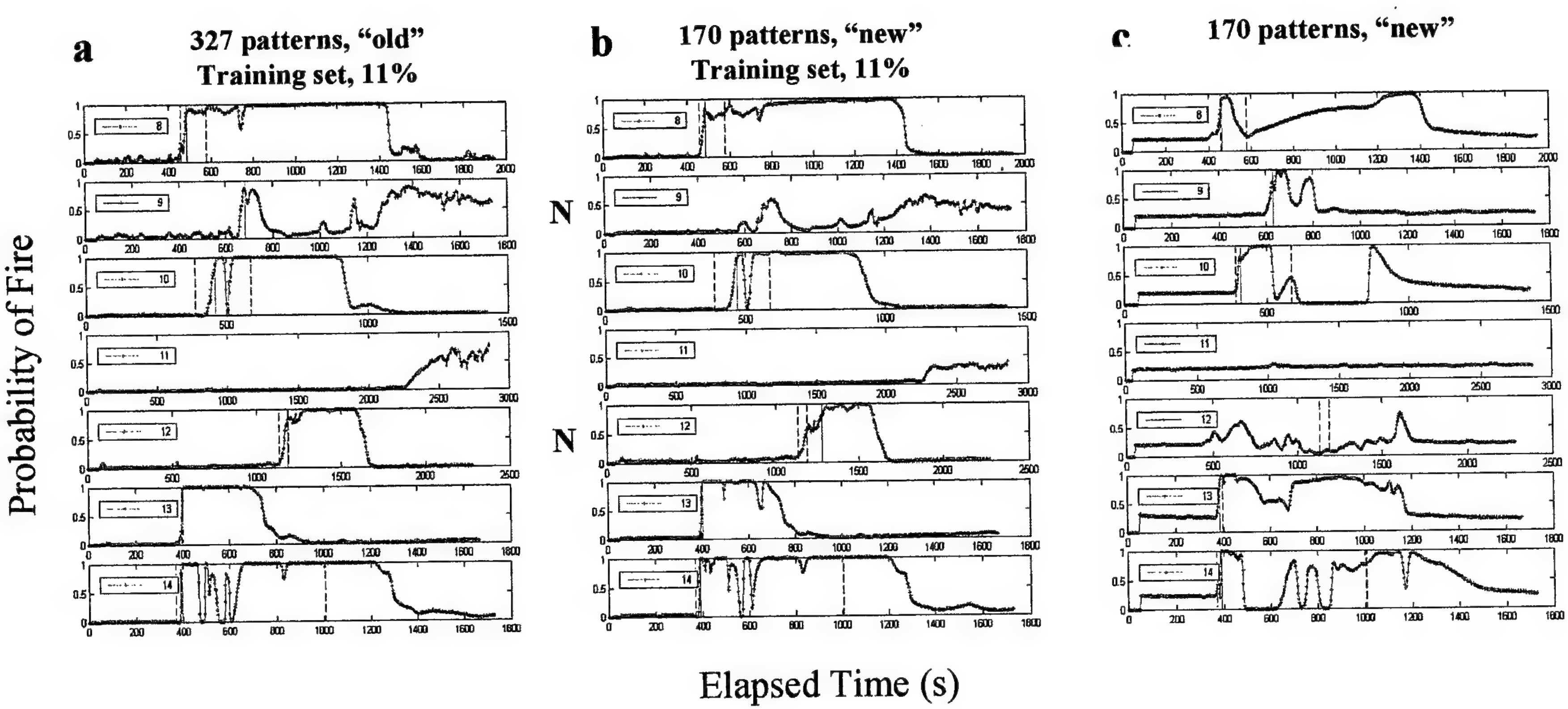

Figure 4. Probability versus elapsed time for Test Series 1 experiments 8-14 (refer to Table 1 for test descriptions) comparing a) the old, 327 pattern training set, generated at the 11 % alarm time, with prototype 1, b) the new, 170 pattern training set, generated at the 11 % alarm time, with prototype 1, and c) the sensor combination [R-ion, CO CO_2] using the new training set generated at the 0.82 % alarm time. Nuisance experiments are denoted with an "N". Solid green line = PNN alarm time, dashed red line = COTS Ion alarm time, and dashed black line = COTS Photo alarm time.

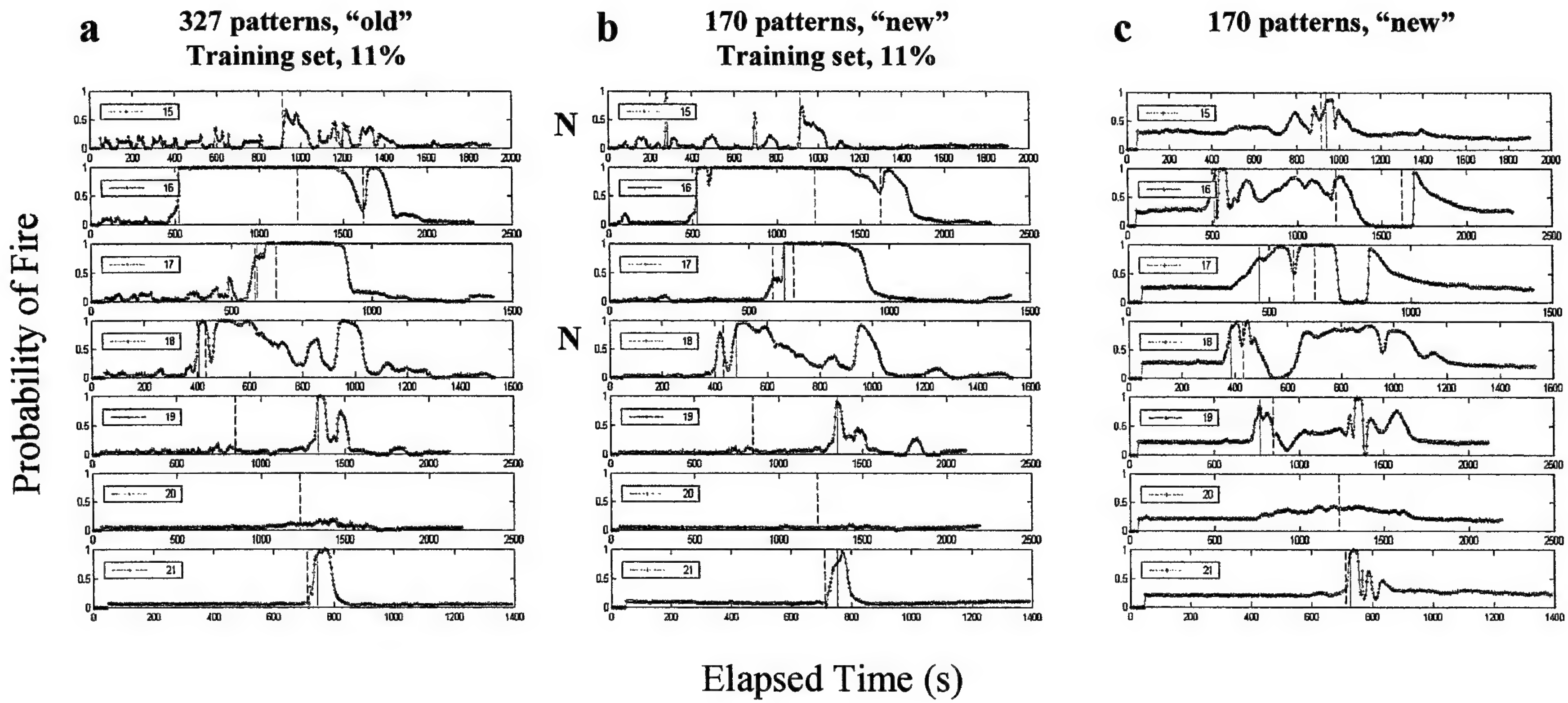

Figure 5. Probability versus elapsed time for Test Series 1 experiments 15-21 (refer to Table 1 for test descriptions) comparing a) the old, 327 pattern training set, generated at the 11 % alarm time, with prototype 1, b) the new, 170 pattern training set, generated at the 11 % alarm time, with prototype 1, and c) the sensor combination [R-ion, CO CO_2] using the new training set generated at the 0.82 % alarm time. Nuisance experiments are denoted with an “N”. Solid green line = PNN alarm time, dashed red line = COTS Ion alarm time, and dashed black line = COTS Photo alarm time.

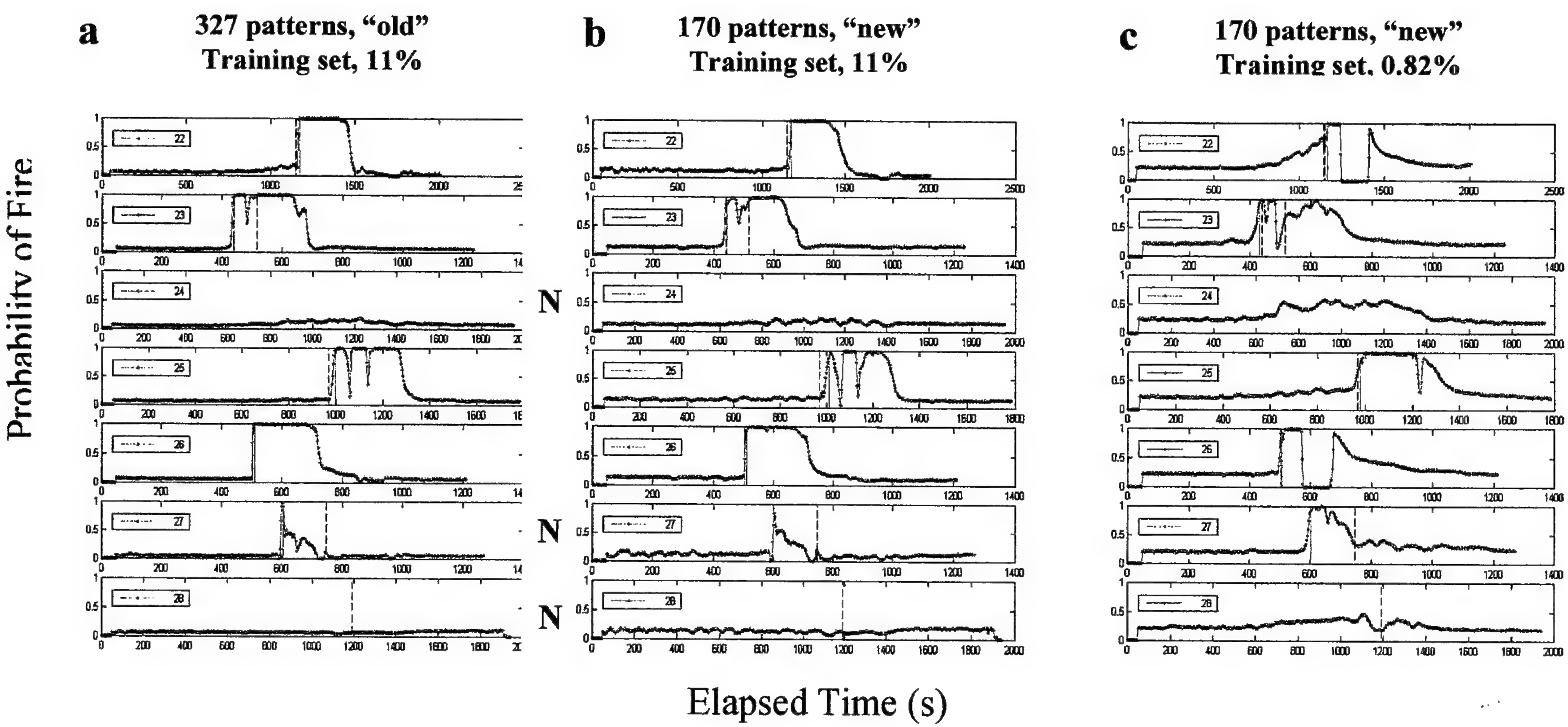

Figure 6. Probability versus elapsed time for Test Series 1 experiments 22-29 (refer to Table 1 for test descriptions) comparing a) the old, 327 pattern training set, generated at the 11 % alarm time, with prototype 1, b) the new, 170 pattern training set, generated at the 11 % alarm time, with prototype 1, and c) the sensor combination [R-ion, CO CO_2] using the new training set generated at the 0.82 % alarm time. Nuisance experiments are denoted with an "N". Solid green line = PNN alarm time, dashed red line = COTS Ion alarm time, and dashed black line = COTS Photo alarm time.

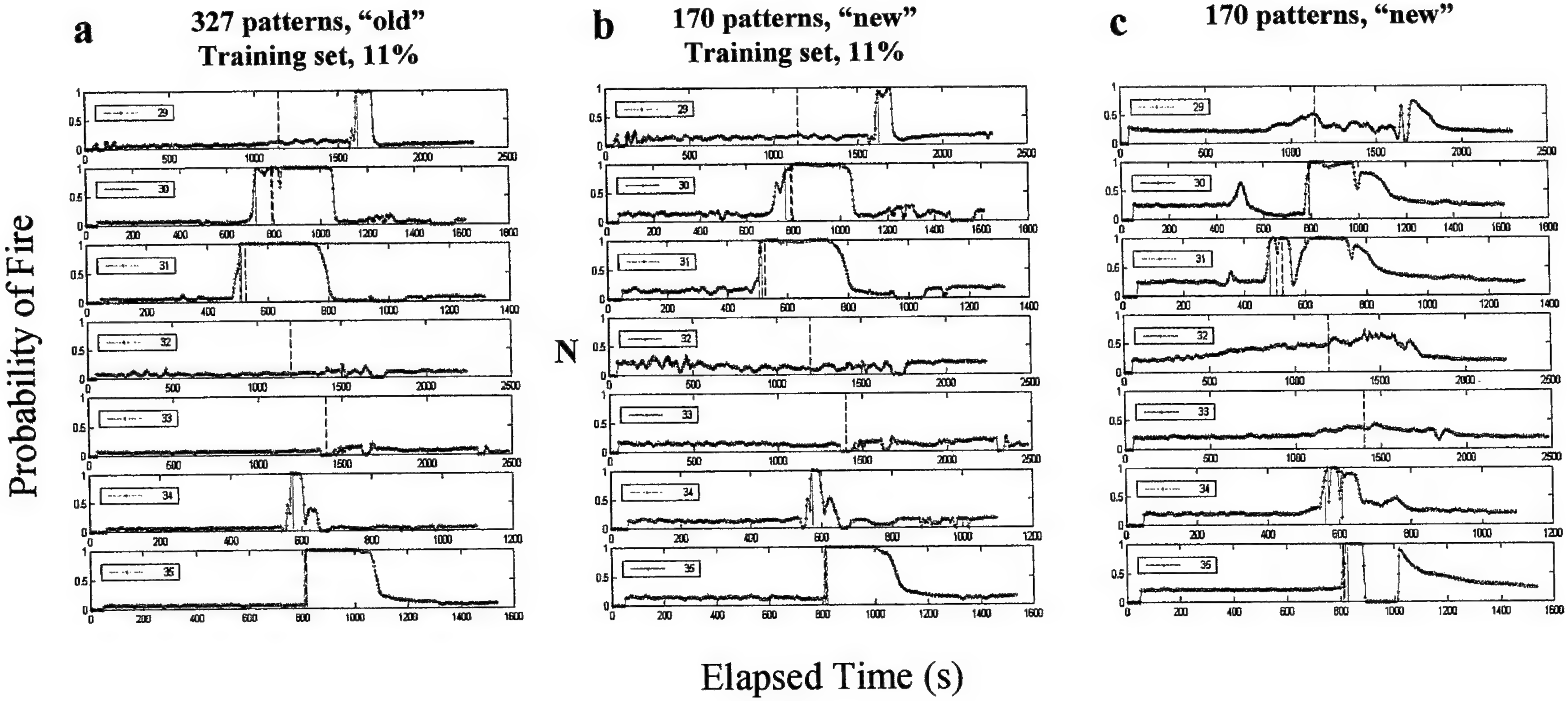

Figure 7. Probability versus elapsed time for Test Series 1 experiments 29-35 (refer to Table 1 for test descriptions) comparing a) the old, 327 pattern training set, generated at the 11 % alarm time, with prototype 1, b) the new, 170 pattern training set, generated at the 11 % alarm time, with prototype 1, and c) the sensor combination [R-ion, CO CO_2] using the new training set generated at the 0.82 % alarm time. Nuisance experiments are denoted with an "N". Solid green line = PNN alarm time, dashed red line = COTS Ion alarm time, and dashed black line = COTS Photo alarm time.

What is immediately noticeable and different with the sensor combination R-ION CO and CO_2 given in Figures 3-7 c is the elevated probability baseline, around 0.3. One would expect that an elevated baseline might lead to a greater number of false alarms due to the proximity to the alarm level and the same rate of increase. However, this is not the case, while the baseline probability level is higher, the rise to alarm level is similar to that of prototypes 1 and 2. Of notable interest in Figures 3 – 7 c is the occasional drop out in the probability plots. The cause of these dropouts is shown in Figure 8 where the processed patterns from the three sensors in the array have been plotted versus elapsed time. In each case, the maximum value has reached a plateau. Aside from missing data, this by itself is not a fatal flaw. However, when these data are presented to a PNN using training data that contains curves that do not reach a plateau, then calculation problems arise. In the calculation of the distance matrix in the PNN, the values of the prediction patterns get subtracted from all patterns in the training set. Significant deviations from the training data range cause instabilities in the distance matrix calculation and subsequent probability calculation. Investigations into the failure modes of the PNN are being undertaken and initial studies are presented in Section 7 of this report.

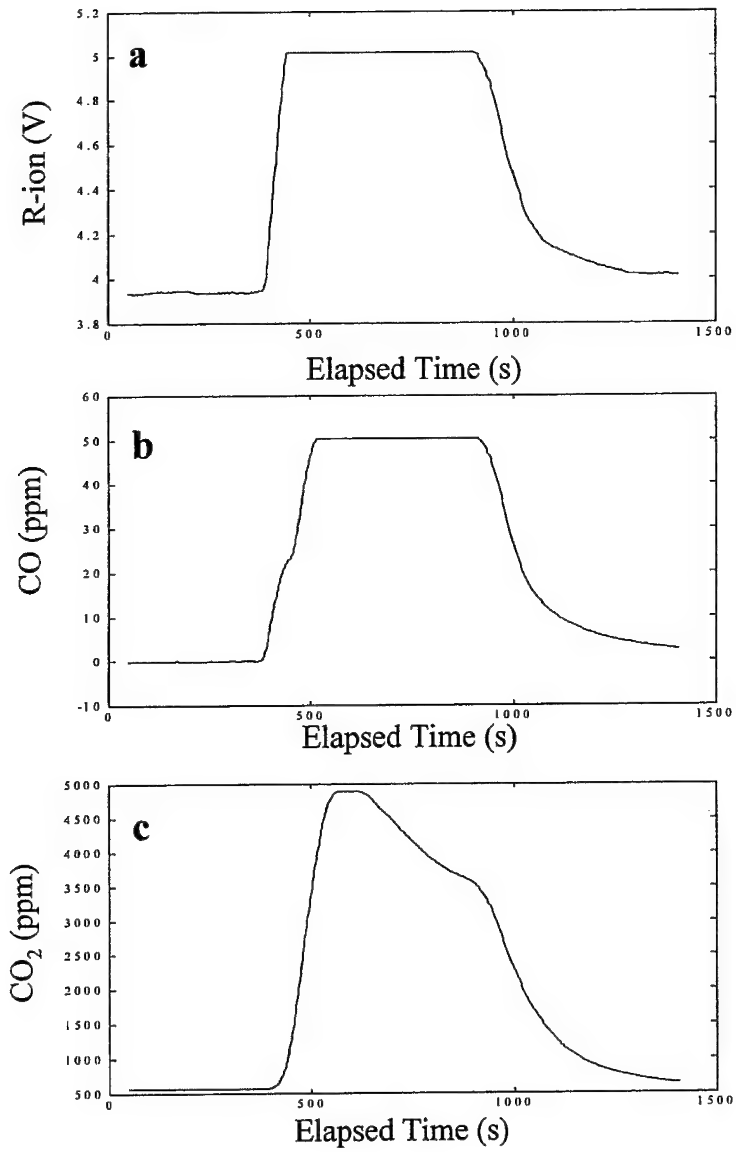

Figure 8. Unscaled / corrected sensor patterns from experiment 3, oily rag, paper in trashcan, for a) R-ION, b) CO, and c) CO_2

7. Basic PNN Simulations and Testing

A variety of simulations and tests were performed to evaluate several data scenarios that are known to have occurred during testing to date. Additional scenarios were investigated to help elucidate PNN function, performance, and failure in this real-time fire classification algorithm. Scenarios include increased noise on a single channel while maintaining the integrity of the other channels, increasing the noise on all channels simultaneously, single sensor dropout (sensor value = 0), and single sensor erroneous values (values set to a negative number or much higher than possible). The data used for

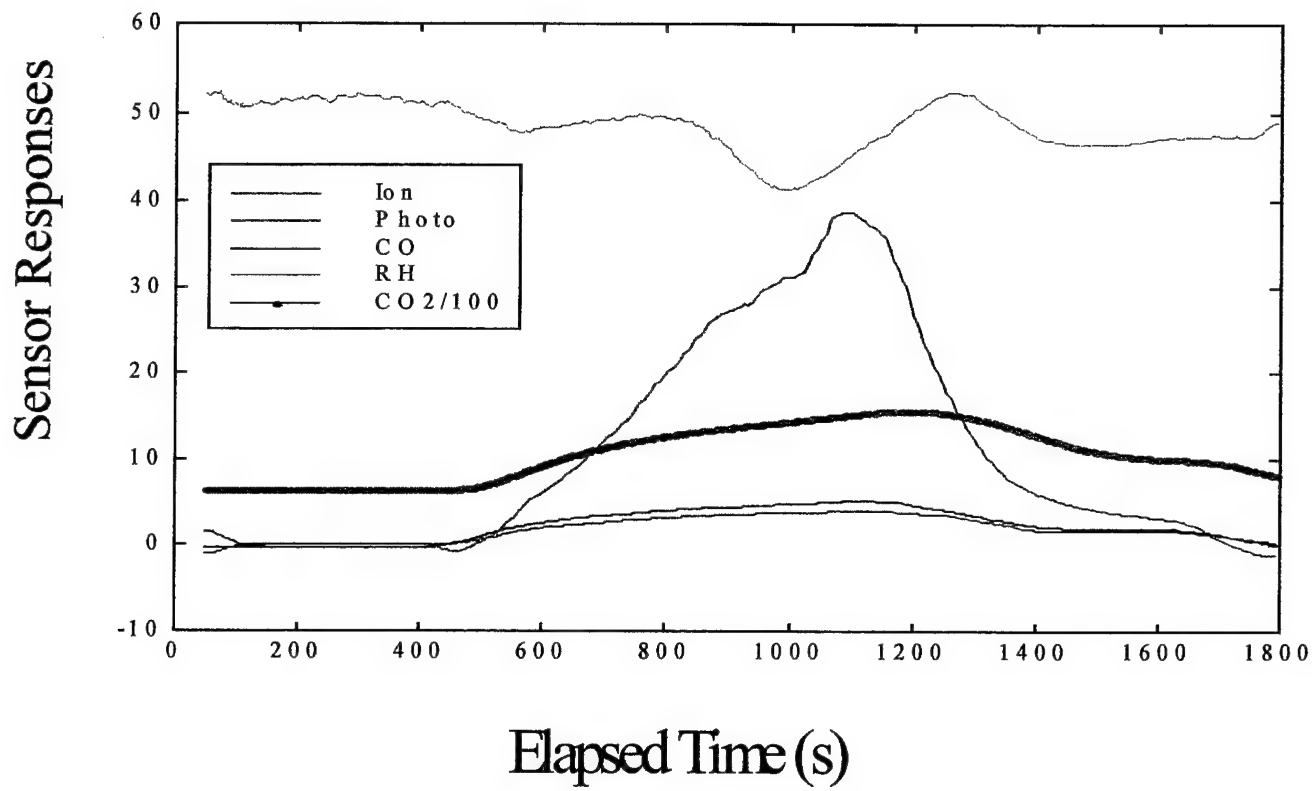

Figure 9. Sensor responses for heptane test EWFD_001 in Test Series 1

testing was the first heptane experiment performed in Test Series 1, shown in Figure 9. An increase in all sensors except % relative humidity (RH), plotted in Figure 9, can be seen at approximately 500 seconds. The data plotted are the patterns generated during the playback of the experiment. This means that each point was generated using a 25-point average, thus smoothing the data. The effect of noise added to the data was investigated by adding noise to the RH sensor alone, while maintaining the integrity of the others. This is a known situation, which has occurred on several sensors, for a variety of reasons, throughout the course of the

Test Series. Plots of the raw humidity data before PNN analysis are shown in Figure 10. They show the low signal to noise ratios (S/N) that were used in order to induce failure in the PNN. The algorithm-processed patterns and the resulting PNN probabilities are shown in Figure 11. The S/N values are markedly improved due to the 25-point averaging performed by the PNN algorithm, as reflected in the S/N values for the processed patterns, and this has a significant effect on the probabilities.

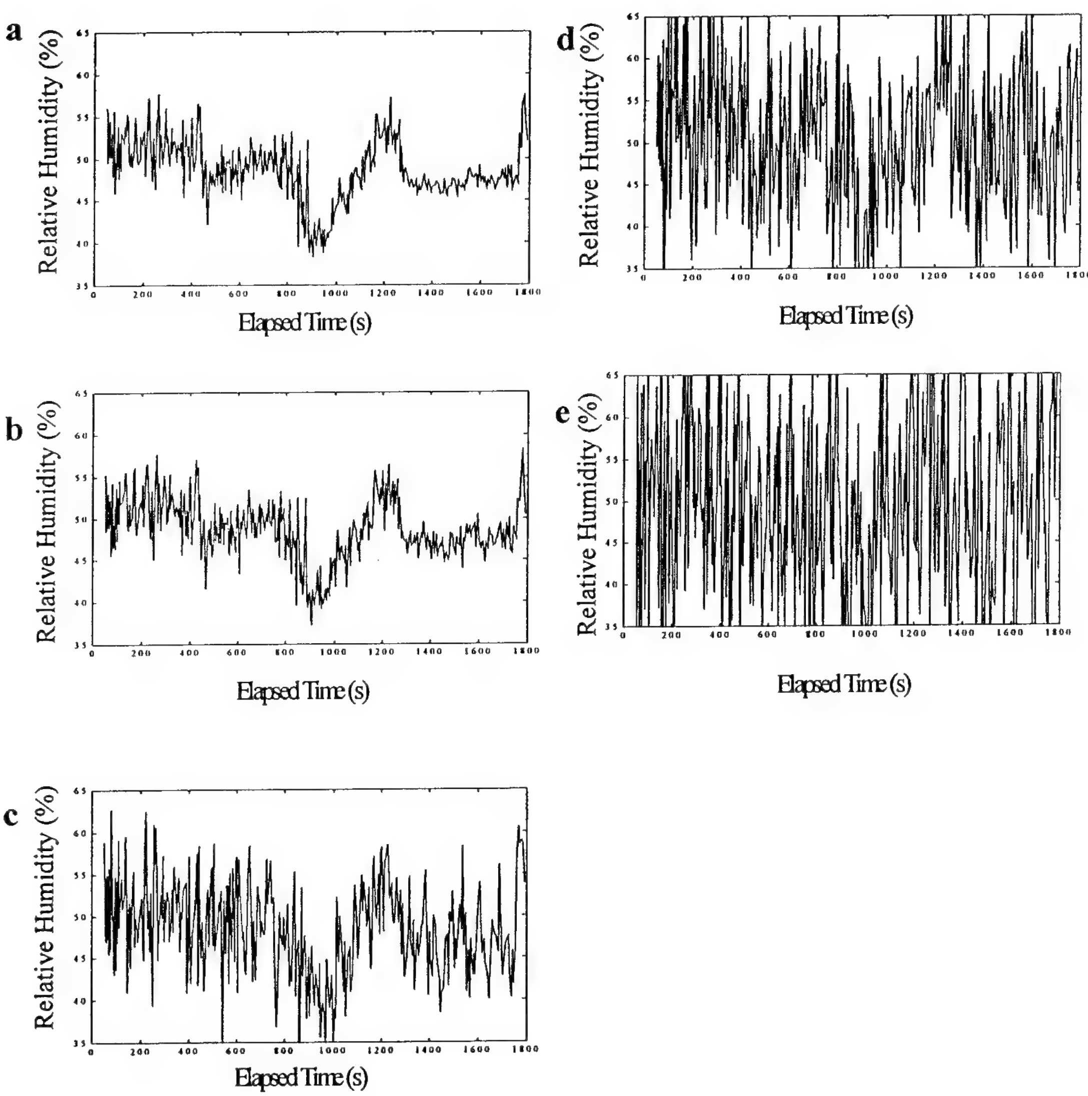

Figure 10. Raw % RH signals as a function of elapsed time during heptane experiment EWFD_001. A) S/N = 18, b) S/N = 17, c) S/N = 9, d) S/N = 5, e) S/N = 4.

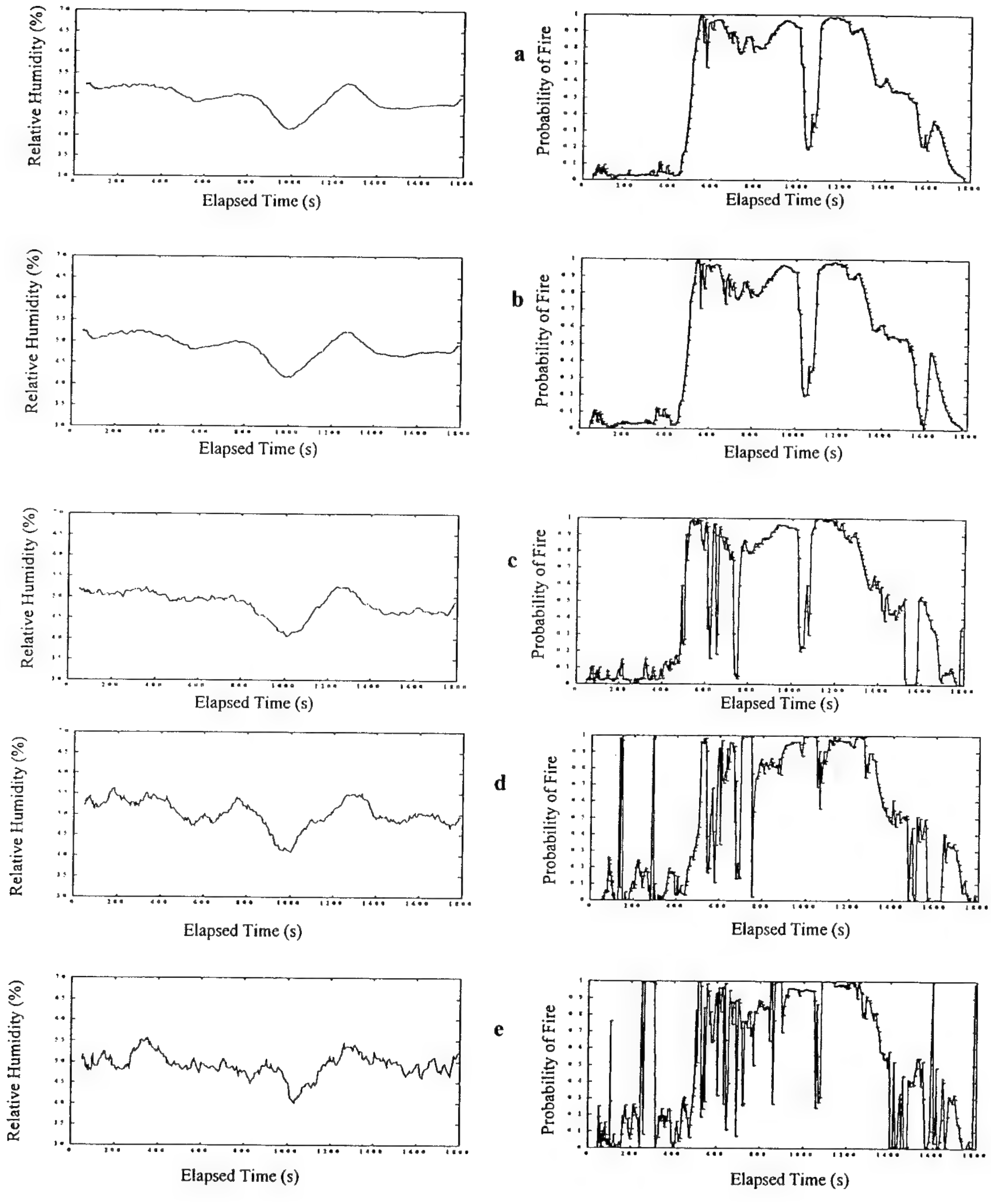

Figure 11. Processed relative humidity (%) patterns versus elapsed time and probability of fire versus elapsed time during heptane fire EWFD_001 for a) original data, S/N = 222, b) noise added to data, S/N = 182, c) noise added to data, S/N = 108, d) noise added to data, S/N = 64, e) noise added to data, S/N = 26.

However, the PNN still reached a failure condition at a raw S/N level of 5, Figure 10 and 11 d. Failure is defined as the PNN predicting alarm states during background measurement before fire ignition coupled with increased instability and noise in all of the probabilities. Additional failure tests were performed and the resulting probabilities are shown in Figure 12. The first plot, a, is the original data

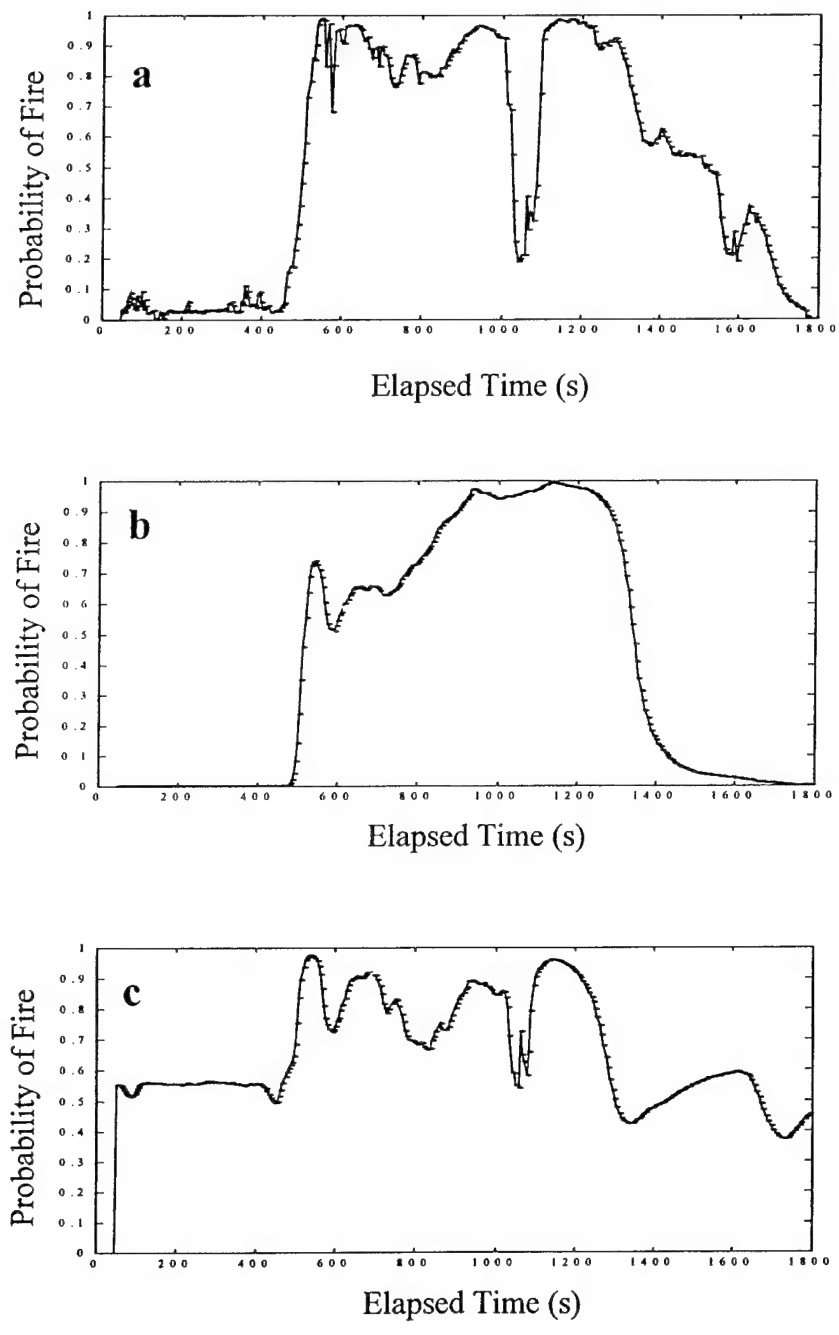

Figure 12. PNN failure tests where % RH was set at a) original data for comparison, b) flat-line zero, c) flat-line 100%.

for comparison and Figure 12 b is the probability result when the %RH sensor drops out to zero during an experiment. The remaining sensors do a good job of predicting reasonable probabilities; The PNN does not completely fail with the loss of this sensor. While it still produces reasonable probabilities, the alarm time would be significantly later (approximately 850 s) than the prototype with a properly functioning %RH sensor (approximately 550 s). The flat-line % RH data, by definition, has no noise. The fact that the resulting probabilities contain very little noise compared with Figure 12 a, the probabilities resulting from the original data indicates that the noise in the %RH sensor was the source of instability in the PNN probabilities. In addition, the reduced probabilities that begin at approximately 1000 s are not present in Figure 12 b. The effect of setting all the %RH values to 100% is shown in Figure 12 c. The elevated %RH values at the beginning of the experiment cause the PNN probabilities to be high, and the resulting variations are similarly skewed.

8. Conclusions

This test series demonstrated an early warning fire detection system consisting of a sensor array and a PNN operated in real time on the ex-USS SHADWELL. During the test, in real time, the prototypes produced a fire classification rate of 89% for prototype 1 and 85% for prototype 2. The nuisance classification rates were 44% for both prototype 1 and 2. The results for fire detection exceed those for the COTS units used onboard the SHADWELL in both classification and speed. The nuisance detection results obtained in real-time were poorer than that achieved by the COTS units. Post-test optimization produced nuisance results using a new training data set (67% and 56% for prototype 1 and 2 respectively) that exceeded the COTS ion unit performance (56%) and matched the COTS photo unit performance (67%), while achieving better fire classification. There were several sensor combinations other than the prototype 1 and 2 that produced nearly equivalent results. However, prototype 1 with the new training set produced the best result with respect to nuisance rejection (67% correct, equivalent to COTS photo detector and better than COTS ion detector, 56% correct).

Additional studies were performed that examined the effects of noise and erroneous sensor responses (zero drop-out, humidity > 100%, excessive noise). The results of these experiments indicate that the PNN is surprisingly robust to variations in sensor performance. Unacceptable performance (unstable probabilities which produce erroneous alarms) was reached in humidity noise testing at S/N = 5, which is reasonably low. Sensor drop out (flat-line zero) during an experiment did not have the catastrophic effect that one might have expected. The remaining sensors and the PNN are capable of producing reasonable results. The main observed effect of sensor drop out was a small increase in alarm time combined with a magnitude reduction in later probabilities. These effects are entirely acceptable for a continuous early warning fire detection system until such time that the problem can be corrected. Future reports will more fully investigate the effects of training sets and preprocessing methods on the optimization of the early warning fire detection system.

9. References

1. D.T. Gottuk, and F.W. Williams, "Multi-Criteria Fire Detection: A Review of the State-of-the-Art," NRL Ltr Rpt Ser 6180/0472, September 10, 1998.

2. H. W. Carhart, T. A. Toomey, and F. W. Williams, " The ex-USS SHADWELL Full-Scale Fire Research and Test Ship", NRL Memorandum Report 6074, October 1987, Reissue, September, 1992.

3. D. T. Gottuk, S. A. Hill, C. F. Schemel, B. D. Strehlen, R. E. Shaffer, S. L. Rose-Pehrsson, P. A. Tatem, and F. W. Williams, "Identification of Fire Signatures for Shipboard Multi-criteria Fire Detection Systems", NRL Memorandum Report, NRL/MR/6180—99-8386, June 18, 1999.

4. R. E. Shaffer, S. L. Rose-Pehrsson, Colin Barry, Daniel T. Gottuk, Frederick W. Williams, "Development of an Early Warning Multi-Criteria Fire Detection System: Analysis of Transient Fire Signatures Using a Probabilistic Neural Network", NRL Memorandum Report, NRL/MR/6110—00-8429, February 16, 2000.

5. S.L Rose-Pehrsson, R. E. Shaffer, S. J Hart, F. W. Williams, D. T. Gottuk, S. A. Hill, and B. D. Strehlen, "Multi-Criteria Fire Detection Systems using a Probabilistic Neural Network", Sensors and Actuators, B, in press.

6 . J. T. Wong, D. T. Gottuk, R. E. Shaffer, S. L. Rose-Pehrsson, S. Hart, P. A. Tatem, and F. W. Williams, "Results of Multi-Criteria Fire Detection System Tests", NRL Memorandum Report, NRL/MR/6180—00-8452, May 22, 2000.

7. S. L. Rose-Pehrsson, S. J. Hart, R. E. Shaffer, J. T. Wong, D. T. Gottuk, P. A. Tatem, and F. W. Williams, "Analysis of Multi-Criteria Fire Detection Systems Results for Test Series 1," NRL Letter Report, NLR Ltr Rpt Ser 6110/116, 16 May 2000.

8 . M.T. Wright, D. T. Gottuk, J. T. Wong, S. L. Rose-Pehrsson, S. J. Hart, P. A. Tatem, and F. W. Williams, "Prototype Early Warning Fire Detection System: Test Series 1 Results," NRL Letter Report, NLR Ltr Rpt Ser 618/0163A:FWW, 27 April, 2000.

9. R. E. Shaffer, S. L. Rose-Pehrsson, R. A. McGill, "A Comparison Study of Chemical Sensor Array Pattern Recognition Algorithms", Anal. Chim. Acta, 384, 1999, 305-317.

10. R. E. Shaffer, S. L. Rose-Pehrsson, R. A. McGill, "Probabilistic Neural Networks for Chemical Sensor Array Pattern Recognition: Comparison Studies, Improvements and Automated Outlier Rejection", Naval Research Laboratory Formal Report 6110-98-9879, 1998.

11. R. E. Shaffer, S. L. Rose-Pehrsson, "Improved Probabilistic Neural Network Algorithm for Chemical Sensor Array Pattern Recognition", *Analytical Chemistry*, 1999, 71(9), p. 4263-4271.

12. T. Masters, *Advanced Algorithms for Neural Networks A C++ Sourcebook*, Academic Press, New York, 1995.

13. S. A., Hill, A. Peters, S. Tweedie, and F. W. Williams, "Results of Commercial off the Shelf Advanced Fire and Smoke Sensor System Tests on ex-USS SHADWELL, Phase II, NRL Ltr Rpt 6180/0192, 17 April, 1998.

Appendix A. Real time PNN Matlab code

```
Function
[w,p,pattern,alarm_state,data_history,buffer_data,alarm_history] =
rtpnncode2lv(Xcurrent,bckgrd,data_history,buffer_data,alarm_history,tra
in,tinfo,sigma,alarmprob)
% RTpnncode2 - Code written to be incorporated into Labview for use
during real-time fire testing on the ex-USS SHADWELL
%
% Version 0.1 - 1/11/00 - Sean J.  Hart
% Version 0.2 - 1/20/00 - SJH - Modified averaging, added alarm
probability and fire criterion for alarms
% Version 0.3 - 1/28/00 - SJH - Removed redundant variable data_average
from input and output lists
% Version 0.4 - 1/31/00 - SJH - Modified code to use preprocess type 3
(engineering units and no background subtraction with selection code 8
(mag/slope calc)
% Version 0.5 - 1/31/00 - SJH - Removed redundant variables buffer size
and average size as they are no longer needed

% Outputs-
% w:                 class winner determined by PNN
% p:                 PNN probability
% pattern:           processed sensor values used by the PNN
% alarm_state:       current alarm condition
% data_history:      a buffer of data points from which to calculate
baselines and slopes
% data_average:      the averaged raw data points from buffer_data
% buffer_data:       a buffer of raw data points from which to calculate
an average raw input
% alarm_history:     a record of alarm conditions
%
% Inputs-
% Xcurrent:          sensor input values
% bckgrd:            backgr supplied by HAI
% data_history:      record of
% alarm_history:     record of alarm states
% train:             Training set patterns
% tinfo:             Training set info
% sigma:             Training kernel width
% alarmprob:         Probability at which to alarm
%
%

[junk,numsensors] = size(Xcurrent);

% *************************** Some constants
******************************************************

slopeflag = 1;
slopelength = 25;
slopelengthbase = 15;
maglength = 10;
maglengthbase = 10;
x = (1:slopelength)';
```

```
if maglength > slopelength
   startcalc = maglength;
else
   startcalc = slopelength;
end

% **** Buffer - average data if collected every second ***************

buffer_data = [buffer_data;Xcurrent]; %add row to end of matrix % Size
of buffer_data defines the number of points averaged
buffer_data = delsamps(buffer_data,1); %remove oldest row in matrix

data_average = mean(buffer_data(:,:));

% ** Background subtraction and conversion calculations **************

 converted_data = data_average;

      [y] = polyval([0.0000034 -0.0004140 0.0171968 -0.2070225
      0.0004794],converted_data(1)); % convert delta mic to %obs/ft
      converted_data(1) = (1-((1-(y/100)).^(3.28)))*100; % convert to
      %/meter
      converted_data(2) = (1-((1-(converted_data(2)/100)).^3.28))*100;

% ****** Data History - average data for baseline / slope calc ********

data_history = [data_history;converted_data]; %add row to end of matrix
- Size of data_history limits the number of points that can be used for
the slope calc
data_history = delsamps(data_history,1); %remove oldest row in matrix

%************** MAG / slope calc ***********************************

for k = 1:numsensors      % each sensor (column)
   testpointsmag = data_history(:,k);
   pattern(:,k) = mean(testpointsmag);
   if slopeflag == 1
      testpointsslope = data_history(:,k);
      temp2 = polyfit(x,testpointsslope,1);
      pattern(:,k+numsensors) = temp2(1);
   end
end

%Select combination of MAG and Slopes to match training set
%pattern = pattern([1 2 3 4 5 6 7 8]);

% ************* Call PNN and trigger alarm state ******************

   rowstoaverage = find(mean(data_history,2)== 0); % makes sure that
zeros are not included in the average at start of data collection

   if (isempty(rowstoaverage) == 1)

      [atrain,mx,stdx] = auto(train); %Autoscale training set - can
remove this set and pass it in each time if speed is required
      [apattern] = scale(pattern,mx,stdx); % Scale prediction pattern
according to training set scaling
```

```
      [w,p] = pnnpred(atrain,tinfo,apattern,sigma,2); % Evaluate
pattern using PNN

   else

      w = 0;
      p = zeros(1,2);

   end

% ******* Probability alarm decision***************************

if  (p(1) > alarmprob) % If PNN probability is greater than cutoff then
set alarm_history
   alarm = 1; % alarm on
   alarm_history = [alarm_history;alarm];
else
   alarm = 0; % alarm on
   alarm_history = [alarm_history;alarm];
end

alarm_history = delsamps(alarm_history,1); %remove oldest row in matrix


% *********** Fire criterion alarm decision*********************

alarmzeros = find(alarm_history == 0); % Find out if the alarm has been
set in any of the last n alarms (size of alarm_history define the check
window)
if (isempty(alarmzeros) == 1) % Set the alarm state on if all elements
of alarm_history have been non-zero (i.e. fire present)
   alarm_state = 1; %Alarm state on - signals a fire
else
   alarm_state = 0; %Alarm state off - signals no fire
end
```